1.

- Identify the characteristics of this situation and specify the logic behind the selection of characteristics:

* The user requirements are somewhat ambiguous and not coherent. They are not easy to understand, especially those who are new to the field. The requirements have low reliability as they are big, complicated and still need further details.
* There are three types of users: pet sitters, managers, and client of businesses. There are two types of requirements: functional and non-functional. They are composed of 5-7 functional requirements for managers and 2 non-functional requirements. For a Web application, the system is not well-defined yet and there are still a lot to figure out.
* The requirements are not highly transparent. The features are not functionally typical for a simple Web app. We can conclude that the requirements are adaptive and may change in the near future.
* It is stated that: “The IT support of PetSitter could assist you in elicitate some missing requirements in the early stage”. It means that if the team has some trouble with jargons, business rules, and comprehending requirements, they can get help from the company’s IT department to clear out any misunderstandings.
* There could be some user involvement in this project as the project needs to validate the user experience to see if the software is well-received and effective or not. Early testing could benefit this project as it heavily relies on its usability and performance.

- Select a model that best fits the characteristics you identified above: For the above reasons, we should choose the Scrum model. It is suitable for this project as it is an adaptive model. Also, this huge complex Web app can be simplified and built through several sprints. Plus, we can plan constant releases to get early feedback from the stakeholders.

2.

- Four functional requirements of manager’s role:

* View a Live Sitter Map, as well as continuous visits, conflict notifications, and a comprehensive scheduling tool.
* Use Pet Journal comments (if enabled), visit notes, or in-system Messages to communicate with employees and clients.
* Track payment and invoice.
* Set up company’s branding and settings.
* Two non-functional requirements of the system:
* The Web application should be compatible with Google Chrome, Safari and Firefox browsers.
* The used language for Web app is English.

3. Two user stories of managers’ role:

- As a manager, I want to view customer feedback of a specific pet sitter so that I can contemplate and assess their performance.

- As a manager, I want to modify a pet sitter’s schedule so that if they are unavailable, I can move them to another date.

4. Story map for the Web app

Payment

Service Management

Account Management

User activities

Track payment

User tasks

Communicate with employees and clients

Manage settings

Login

View service details

View Live Sitter Map

Use Pet Care Journal comments

View invoice detail

Set up company branding

Input user ID and password

Modify an invoice

Add a setting

View visit notes

View conflict notifications

Login by Gmail

View service schedule

Forgot password

Modify a setting

Delete an invoice

Chat with in-system Messages

Log out

5.

a. I totally agree that we should use Agile Software Development to apply to this project. Agile is an adaptive model and has high flexibility. If there are any new requirements, we can add them to the next sprint and discuss them in the daily sprint meeting. The project can be divided into several chunks and developed incrementally. Thus, we can build early releases for the customer and receive instant feedbacks. The team would better engage with the business clients and have further understanding of user requirements. For Agile planning, the team would attend in a get-together to create epics and themes. Under each theme, the members think of user stories for each role, ranging from high priority at the top and low at the bottom. For Agile estimation, we can calculate the individual effort and use Agile estimation techniques (ideal days, relative sizing, etc). For Agile tracking, we implement the Burndown chart to see the work remaining and the progress we have made throughout the days of the sprint. Agile also enables earlier testing than the traditional Waterfall method. Therefore, bugs appear sooner in the developing process, which helps QAs to fix them instantly and reduce technical debts. Furthermore, the stakeholders participate more in the software implementation. This results in higher customer satisfaction as they gain control over the product vision and see its improvement.

b. I would suggest the team to do unit testing, user acceptance testing, system testing, and Continuous Integration. For unit testing, we need to test every single component to make sure they run right before deploying. For acceptance testing, the project will be tested in the end user environment. The users would give the team honest opinions and experience. Then, we discuss if we want to readjust features according to user feedbacks. For system testing, the test is judged upon how well the integrated system interacts with other modules as a whole. Lastly, we can perform Continuous Integration to automate the testing process and shorten the process time. If there are any errors whilst deploying the project, the system will automatically roll back. Otherwise, it will commit successfully.